Lab Cycle 3

1. Sarah bought a new car in 2001 for $24,000. The dollar value of her car changed each year as shown in

the table below.

Value of Sarah’s Car

Year Value

2001 $24,000

2002 $22,500

2003 $19,700

2004 $17,500

2005 $14,500

2006 $10,000

2007 $ 5,800

Represent the following information using a line graph with following style properties

 X- axis - Year

Y –axis - Car Value

 title –Value Depreciation (left Aligned)

 Line Style dash-dot and Line-color should be red

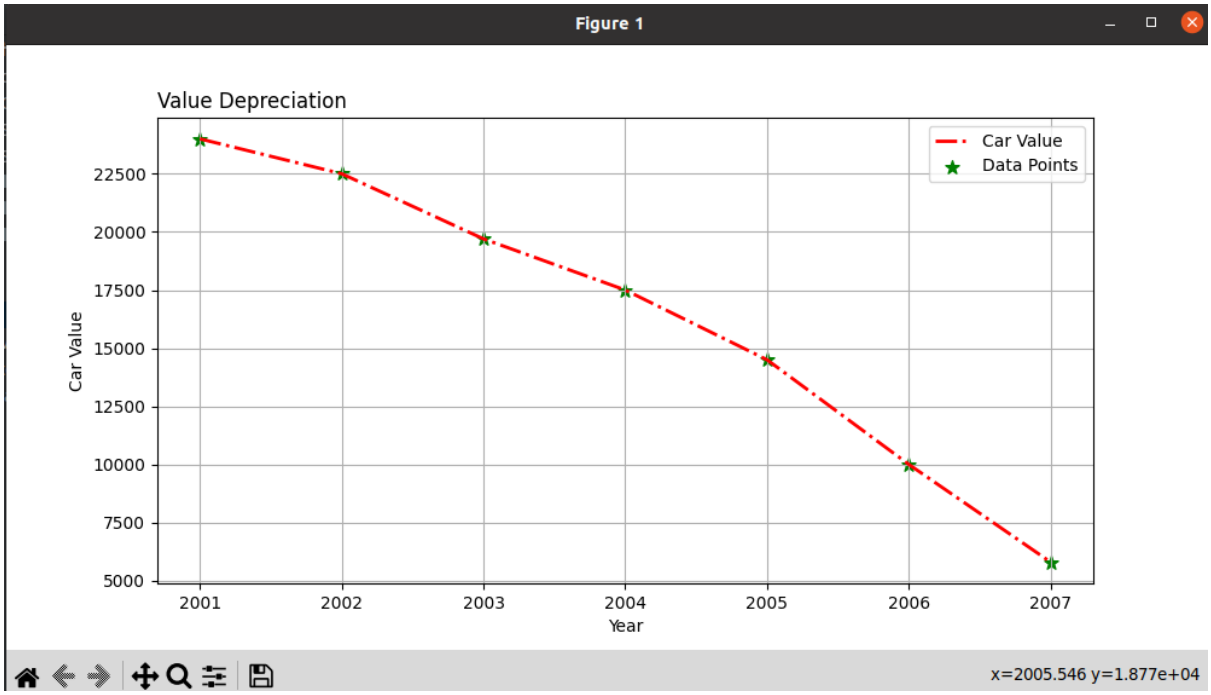
 point using * symbol with green color and size 20

Input:

```
import matplotlib.pyplot as plt

years = [2001, 2002, 2003, 2004, 2005, 2006, 2007]
car_values = [24000, 22500, 19700, 17500, 14500, 10000, 5800]

plt.figure(figsize=(10, 5))
plt.subplot(1, 1, 1)

plt.plot(years, car_values, 'r-.', label='Car Value', linewidth=2)
```

```
plt.scatter(years, car_values, c='green', marker='*', s=70, label='Data Points')

plt.xlabel('Year')
plt.ylabel('Car Value')
plt.title('Value Depreciation', loc='left')

plt.legend()

plt.grid(True)
plt.show()
```

Output:



2. Following table gives the daily sales of the following items in a shop

Day	Mon	Tue	Wed	Thurs	Fri
Drinks	300	450	150	400	650
Food	400	500	350	300	500

Use subplot function to draw the line graphs with grids (color as blue and line style dotted) for the above information as 2 separate graphs in two rows

a) Properties for the Graph 1:

- X label- Days of week
- Y label-Sale of Drinks
- Title-Sales Data1 (right aligned)
- Line –dotted with cyan color
- Points- hexagon shape with color magenta and outline black

b) Properties for the Graph 2:

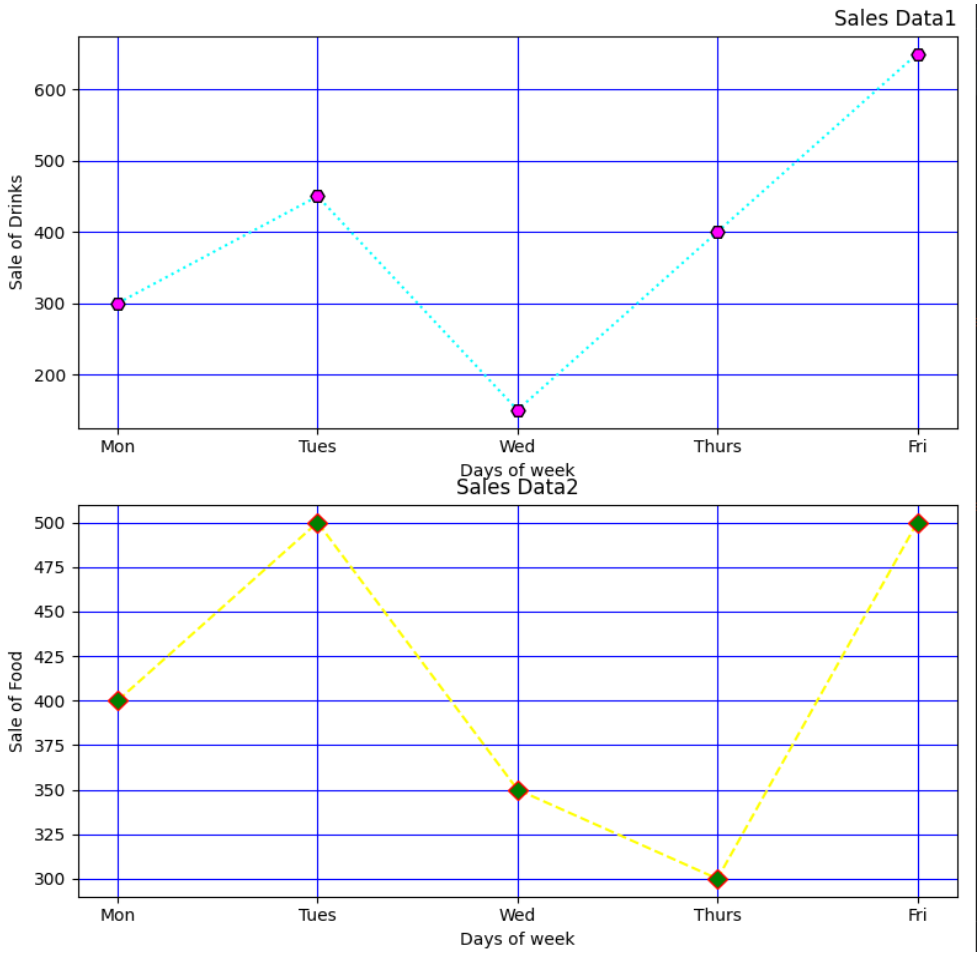
- X label- Days of Week
- Y label-Sale of Food
- Title-Sales Data2 (center aligned)
- Line –dashed with yellow color
- Points- diamond shape with color green and outline red

Input:

```
import matplotlib.pyplot as plt


days = ['Mon', 'Tues', 'Wed', 'Thurs', 'Fri']

sales_drinks = [300, 450, 150, 400, 650]

sales_food = [400, 500, 350, 300, 500]


fig, axes = plt.subplots(2, 1, figsize=(8, 10))


axes[0].plot(days, sales_drinks, linestyle='dotted', color='cyan', marker='H',
markersize=8, markerfacecolor='magenta', markeredgecolor='black')
```

```
axes[0].set_xlabel('Days of week')
axes[0].set_ylabel('Sale of Drinks')
axes[0].set_title('Sales Data1', loc='right')
axes[0].grid(True, color='blue')

axes[1].plot(days, sales_food, linestyle='dashed', color='yellow', marker='D',
markersize=8, markerfacecolor='green', markeredgecolor='red')
axes[1].set_xlabel('Days of week')
axes[1].set_ylabel('Sale of Food')
axes[1].set_title('Sales Data2', loc='center')
axes[1].grid(True, color='blue')
plt.tight_layout()
plt.show()
```

Output:



3.Create scatter plot for the below data:(use Scatter function)

Product	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Affordable Segment	173	153	195	147	120	144	148	109	174	130	172	131
Luxury Segment	189	189	105	112	173	109	151	197	174	145	177	161
Super Luxury Segment	185	185	126	134	196	153	112	133	200	145	167	110

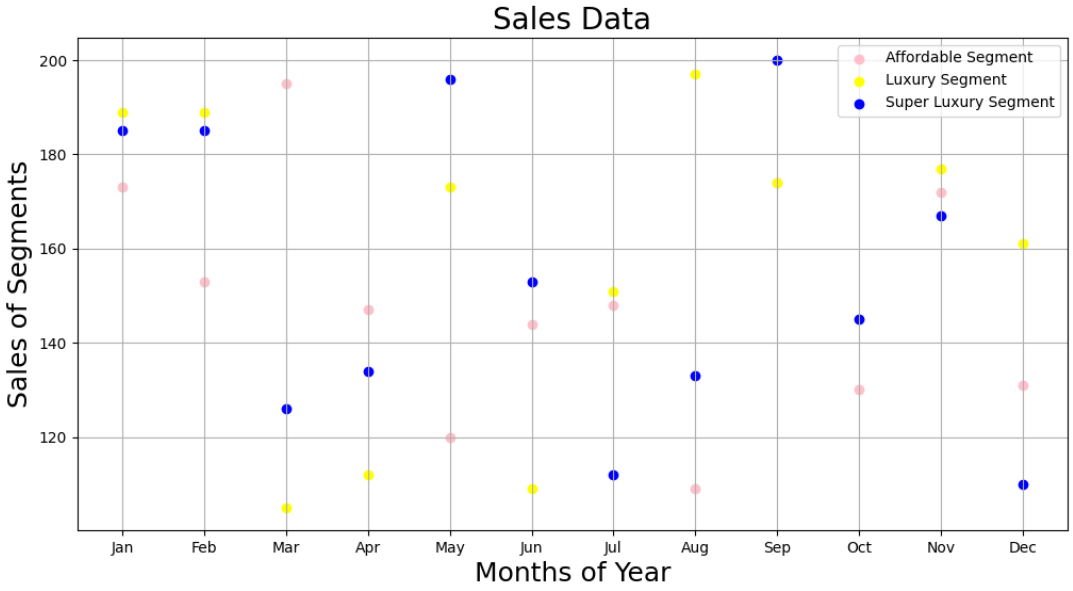
Create scatter plot for each Segment with following properties within one graph

- X Label- Months of Year with font size 18
- Y-Label- Sales of Segments
- Title –Sales Data
- Color for Affordable segment- pink
- Color for Luxury Segment- Yellow
- Color for Super luxury segment-blue

Input:

```
import matplotlib.pyplot as plt

months = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec']
affordable_sales = [173, 153, 195, 147, 120, 144, 148, 109, 174, 130, 172, 131]
luxury_sales = [189, 189, 105, 112, 173, 109, 151, 197, 174, 145, 177, 161]
super_luxury_sales = [185, 185, 126, 134, 196, 153, 112, 133, 200, 145, 167, 110]

plt.figure(figsize=(12, 6))
plt.scatter(months, affordable_sales, color='pink', label='Affordable Segment')
plt.scatter(months, luxury_sales, color='yellow', label='Luxury Segment')
plt.scatter(months, super_luxury_sales, color='blue', label='Super Luxury Segment')

plt.xlabel('Months of Year', fontsize=18)
plt.ylabel('Sales of Segments', fontsize=18)
plt.title('Sales Data', fontsize=20)
plt.legend()

plt.grid()
plt.show()
```

Output:


Sales Data
Affordable Segment
Luxury Segment
Super Luxury Segment
Sales of Segments
200
180
160
140
120
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Months of Year

4. Display the above data using multiline plot (3 different lines in same graph)

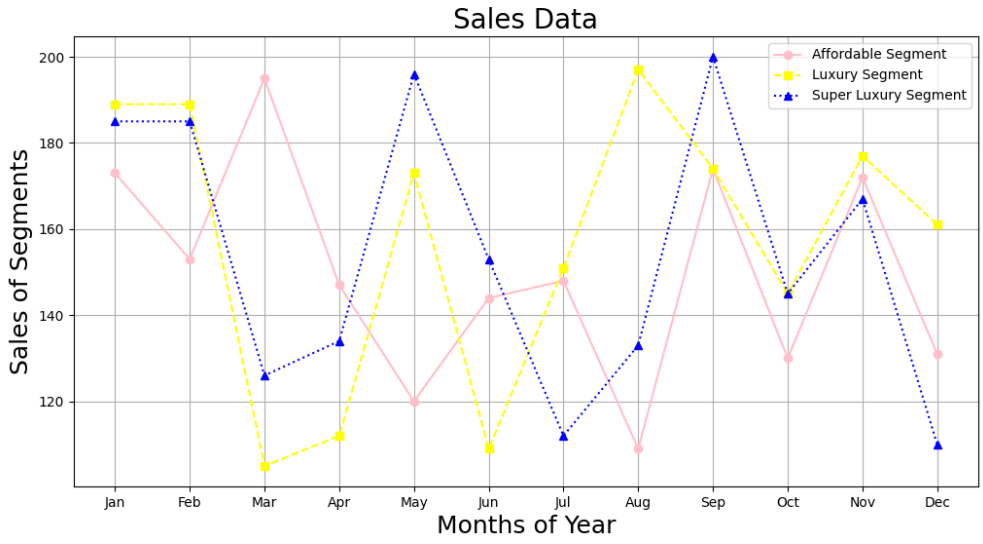
 Display the description of the graph in upper right corner (use legend)

 Use different colors and line styles for 3 different lines

Input:

```
import matplotlib.pyplot as plt

months = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec']
affordable_sales = [173, 153, 195, 147, 120, 144, 148, 109, 174, 130, 172, 131]
luxury_sales = [189, 189, 105, 112, 173, 109, 151, 197, 174, 145, 177, 161]
super_luxury_sales = [185, 185, 126, 134, 196, 153, 112, 133, 200, 145, 167, 110]

plt.figure(figsize=(12, 6))
plt.plot(months, affordable_sales, color='pink', linestyle='-', marker='o',
label='Affordable Segment')
plt.plot(months, luxury_sales, color='yellow', linestyle='--', marker='s', label='Luxury
Segment')
plt.plot(months, super_luxury_sales, color='blue', linestyle=':', marker='^',
label='Super Luxury Segment')

plt.xlabel('Months of Year', fontsize=18)
plt.ylabel('Sales of Segments', fontsize=18)
plt.title('Sales Data', fontsize=20)
plt.legend(loc='upper right')

plt.grid()
plt.show()
```

Output:



5. Create a bar graph with

 X axis -mode of Transport and Y axis ‘frequency’

 Provide appropriate labels and title

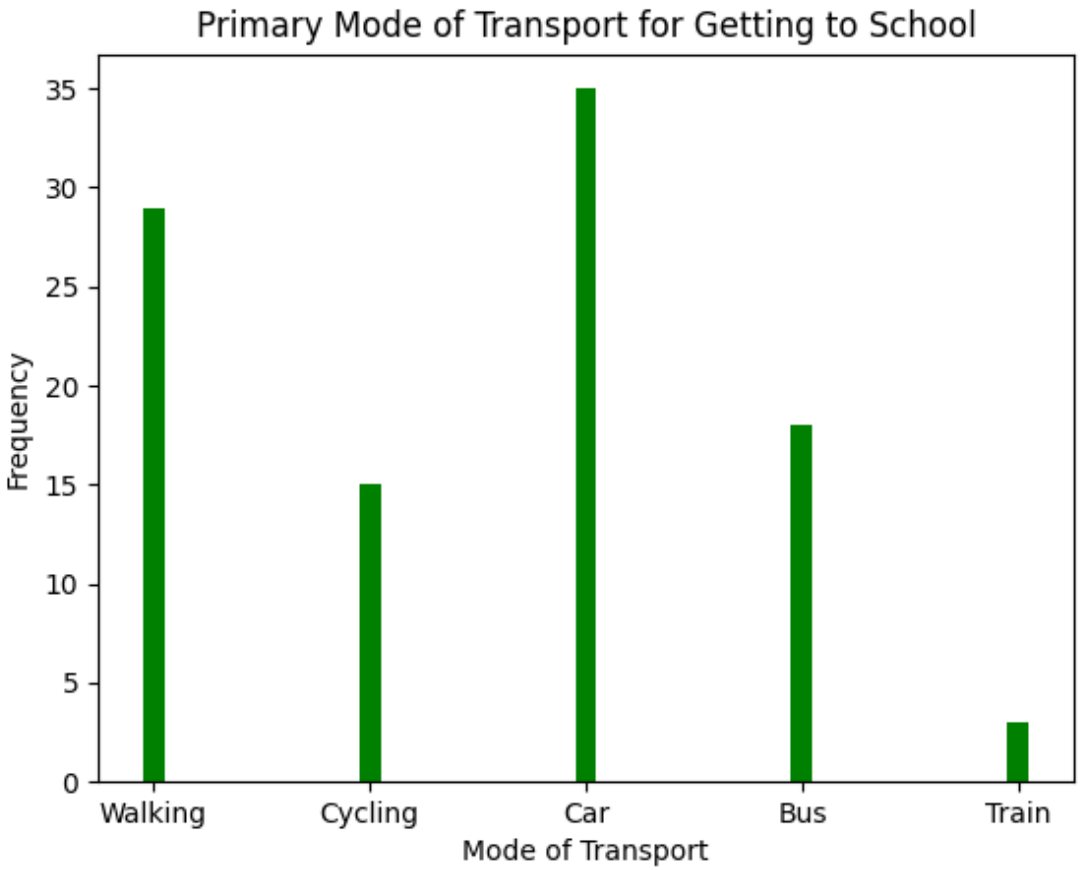
 Width .1, color green

Mode of transport	Frequency
Walking	29
Cycling	15
Car	35
Bus	18
Train	3

Input:

```
import matplotlib.pyplot as plt

modes = ["Walking", "Cycling", "Car", "Bus", "Train"]
frequencies = [29, 15, 35, 18, 3]

width = 0.1

color = "green"

plt.bar(modes, frequencies, width=width, color=color)

plt.xlabel("Mode of Transport")
plt.ylabel("Frequency")
plt.title("Primary Mode of Transport for Getting to School")

plt.show()
```

Output:


Primary Mode of Transport for Getting to School
Frequency
0
5
10
15
20
25
30
35
Walking
Cycling
Car
Bus
Train
Mode of Transport

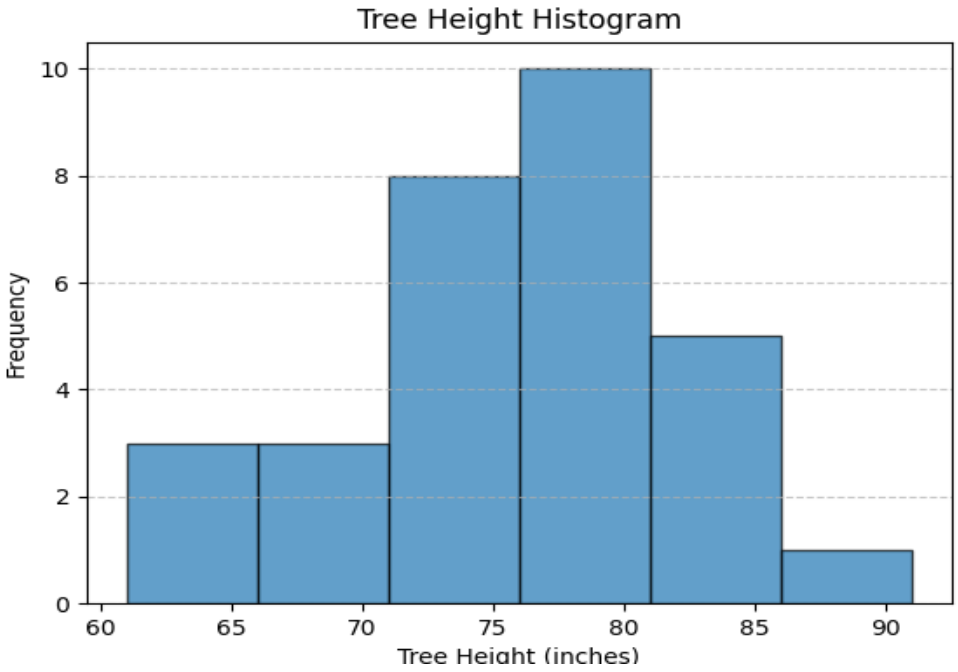
6. We are provided with the height of 30 cherry trees.

The height of the trees (in inches): 61, 63, 64, 66, 68, 69, 71, 71.5, 72, 72.5, 73, 73.5, 74, 74.5, 76, 76.2, 76.5, 77, 77.5, 78, 78.5, 79, 79.2, 80, 81, 82, 83, 84, 85, 87. Create a histogram with a bin size of 5

Input:

```
import numpy as np

import matplotlib.pyplot as plt

tree_heights = np.array([61, 63, 64, 66, 68, 69, 71, 71.5, 72, 72.5, 73, 73.5, 74, 74.5, 76, 76.2, 76.5, 77, 77.5, 78, 78.5, 79, 79.2, 80, 81, 82, 83, 84, 85, 87])

bin_size = 5

hist, bins = np.histogram(tree_heights, bins=np.arange(min(tree_heights), max(tree_heights) + bin_size, bin_size))

plt.hist(tree_heights, bins=bins, edgecolor='black', alpha=0.7)

plt.xlabel('Tree Height (inches)')

plt.ylabel('Frequency')

plt.title('Tree Height Histogram')

plt.grid(axis='y', linestyle='--', alpha=0.7)

plt.show()
```

Output:



7. Using the pandas function read_csv, read the given ‘iris’ data set.

i) Display Shape of the data set.

ii) First 5 and last five rows of data set (head and tail).

iii) Size of dataset.

iv) No. of samples available for each variety.

v) Description of the data set (use describe).

Input:

```
import pandas as pd
print("Name: Athul Ajay")
print("Reg No: SJC22MCA-2017")
print("Batch: 22-24")
print()
data = pd.read_csv('iris.csv')

size = data.size
print("Size = {}".format(size))
print()

print(data.head(5))
print()
print(data.tail(5))
print()

shape = data.shape
print("Shape = {}".format(shape))
print()
occur = data.groupby(['variety']).size()

print('Occurrences of variety',occur)
print()
print(data.describe())
```

Output:

```
/home/sjcet/PycharmProjects/Athul/venv/bin/python /home/sjcet/PycharmProjects/Athul/S3/C3/q7.py
Name: Athul Ajay
Reg No: SJC22MCA-2017
Batch: 22-24

Size = 750

   sepal.length  sepal.width  petal.length  petal.width variety
0           5.1          3.5           1.4          0.2  Setosa
1           4.9          3.0           1.4          0.2  Setosa
2           4.7          3.2           1.3          0.2  Setosa
3           4.6          3.1           1.5          0.2  Setosa
4           5.0          3.6           1.4          0.2  Setosa

     sepal.length  sepal.width  petal.length  petal.width    variety
145           6.7          3.0           5.2          2.3  Virginica
146           6.3          2.5           5.0          1.9  Virginica
147           6.5          3.0           5.2          2.0  Virginica
148           6.2          3.4           5.4          2.3  Virginica
149           5.9          3.0           5.1          1.8  Virginica

Shape = (150, 5)

Occurrences of variety variety
Setosa        50
Versicolor    50
Virginica     50
dtype: int64
```

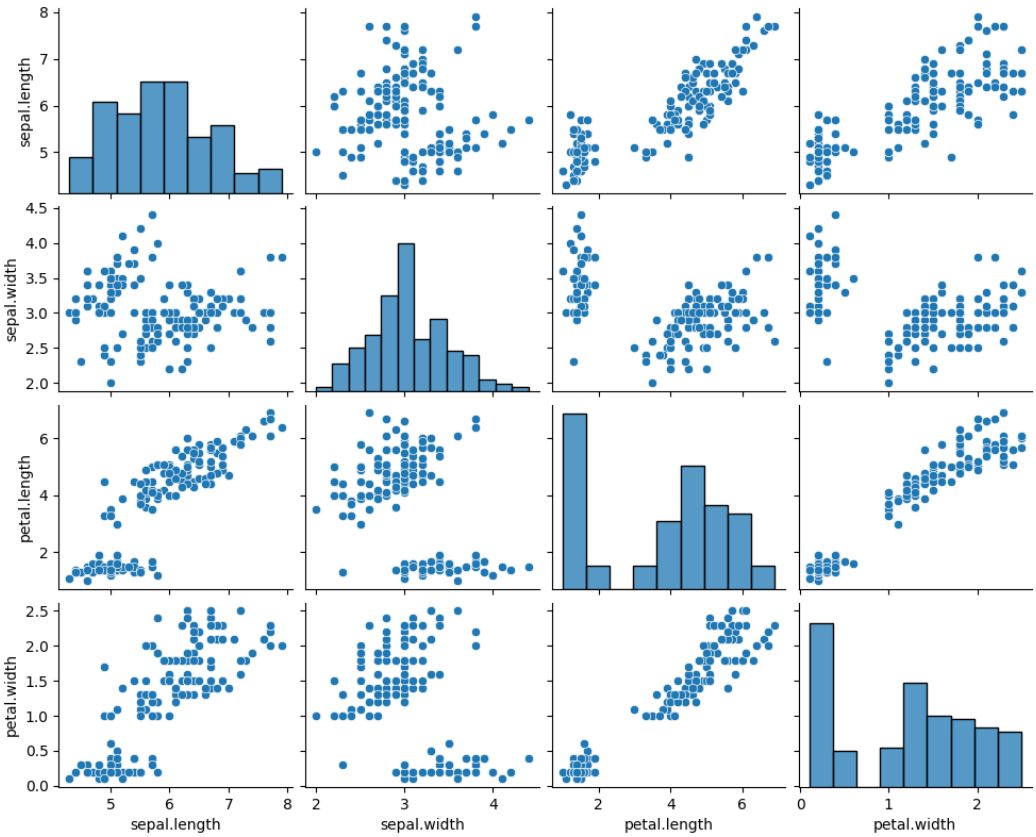
```
       sepal.length  sepal.width  petal.length  petal.width
count    150.000000   150.000000    150.000000   150.000000
mean       5.843333     3.057333      3.758000     1.199333
std        0.828066     0.435866      1.765298     0.762238
min        4.300000     2.000000      1.000000     0.100000
25%        5.100000     2.800000      1.600000     0.300000
50%        5.800000     3.000000      4.350000     1.300000
75%        6.400000     3.300000      5.100000     1.800000
max        7.900000     4.400000      6.900000     2.500000

Process finished with exit code 0
```

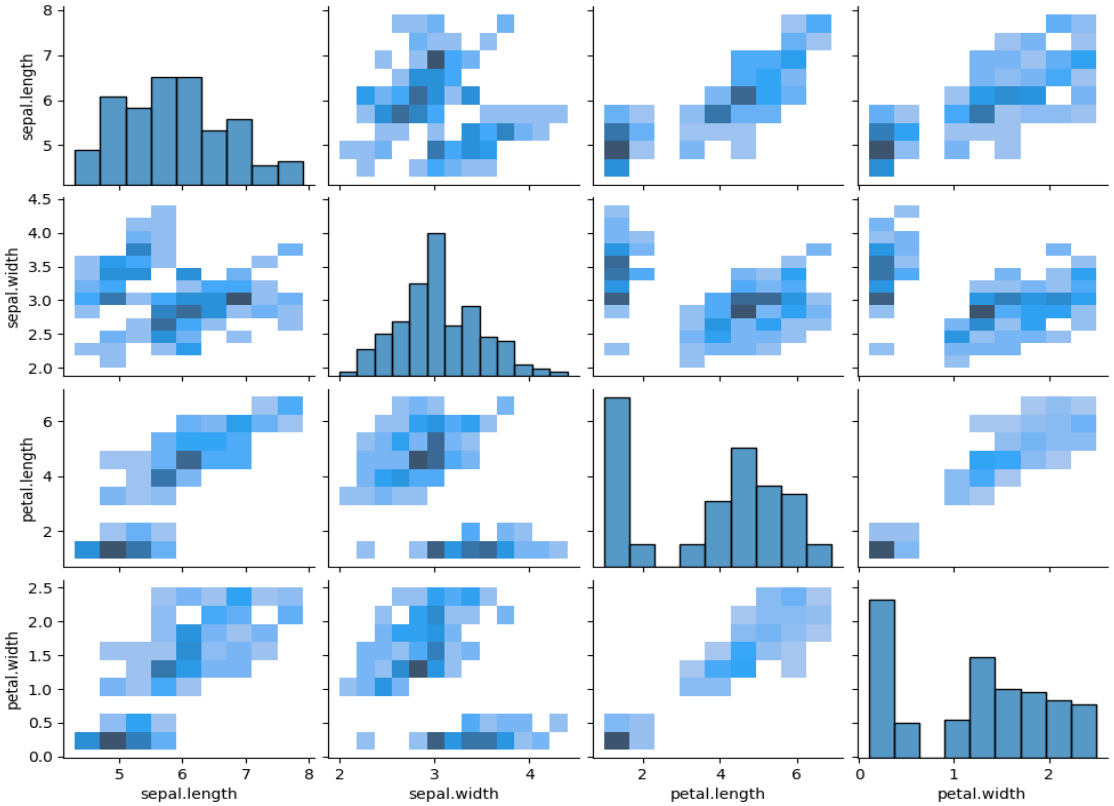
8. Use pairplot() function in seaborn to display pairwise relationships between attributes. Try different kind of plots {‘scatter’, ‘kde’, ‘hist’, ‘reg’} and different kind of markers

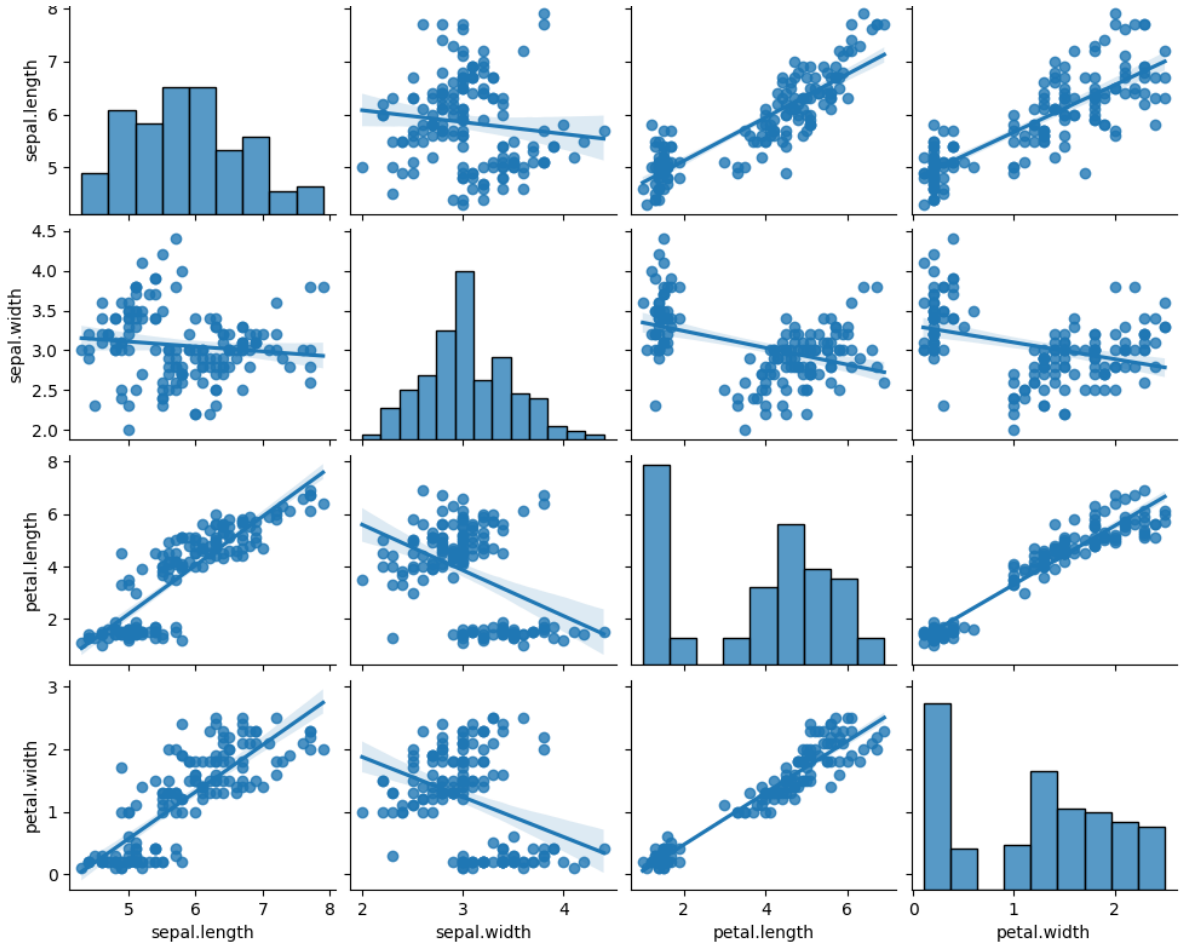
Input:

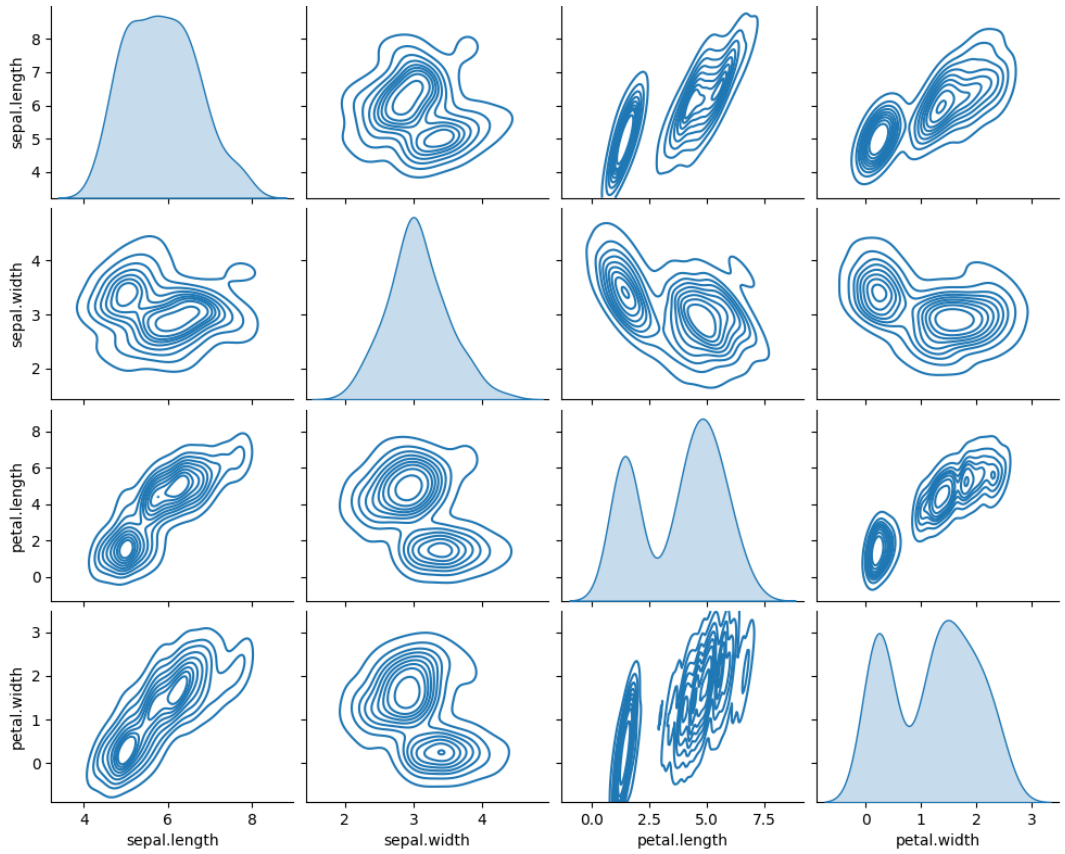
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
data = pd.read_csv('iris.csv')
sns.pairplot(data, kind="scatter")
plt.show()

#sns.pairplot(data, kind="hist")
#plt.show()

#sns.pairplot(data, kind="kde")
#plt.show()

#sns.pairplot(data, kind="reg")
#plt.show()

#sns.pairplot(data, hue="variety", kind="scatter" , markers=["o", "s", "D"])
#plt.show()
```

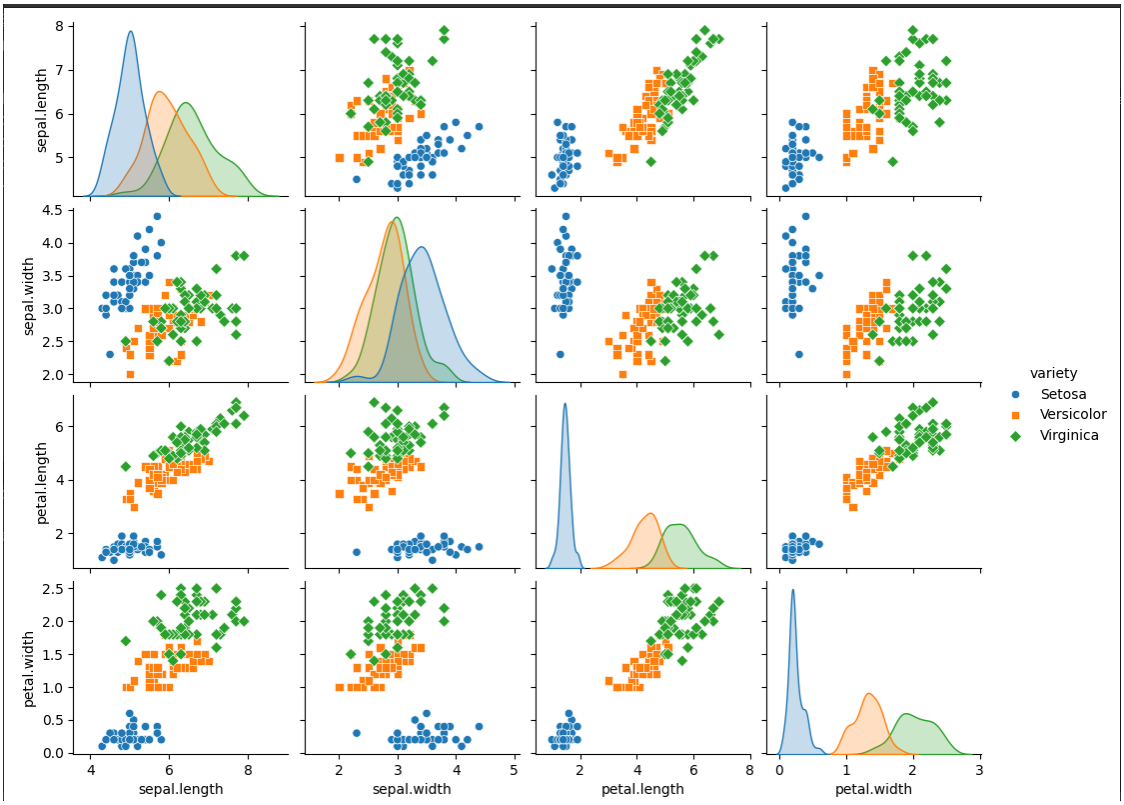
Output:




sepal.length
sepal.width
petal.length
petal.width


sepal.length
sepal.width
petal.length
petal.width


sepal.length
sepal.width
petal.length
petal.width


sepal.length
sepal.width
petal.length
petal.width
variety
Setosa
Versicolor
Virginica

9. Using the iris data set,get familiarize with functions:

1) displot()

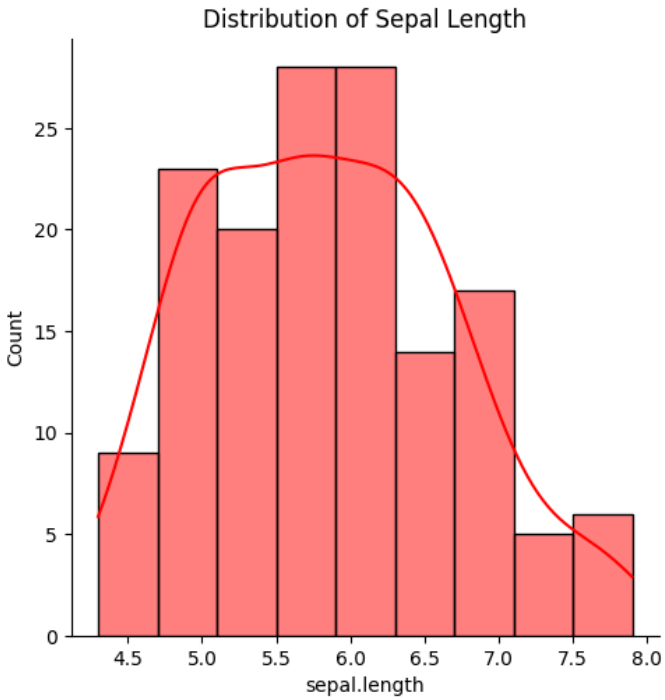
2) histplot()

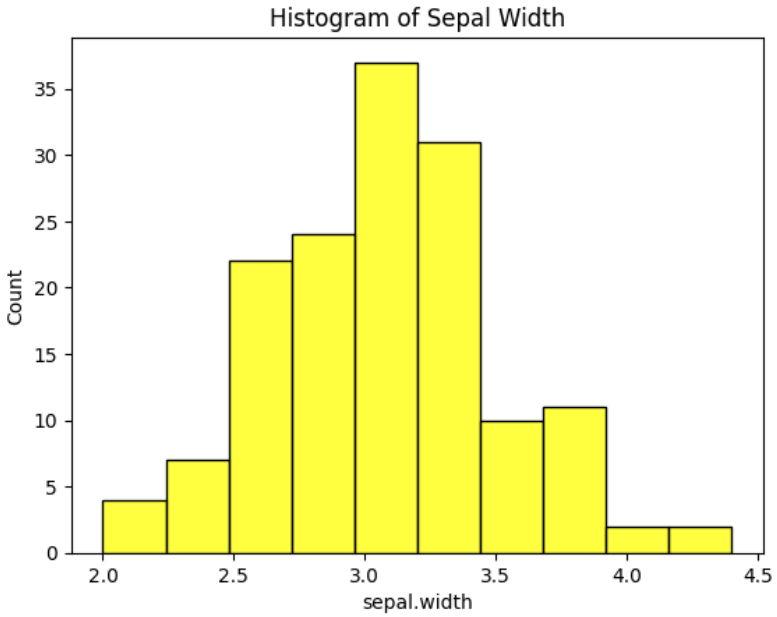
3) relplot()

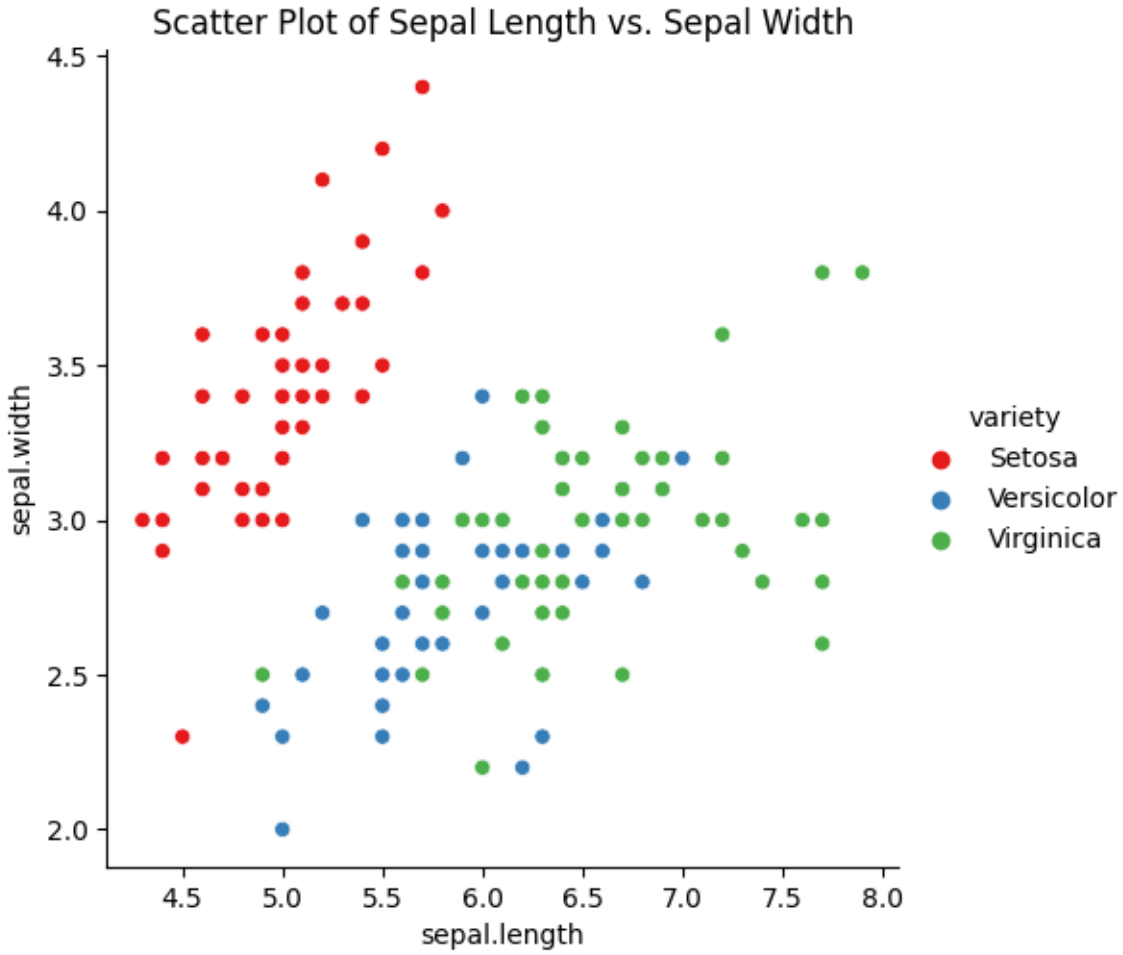
Input:

```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd

iris_data = pd.read_csv("iris.csv")

sns.displot(data=iris_data, x="sepal.length", kde=True, color="red")
plt.title("Distribution of Sepal Length")
plt.show()

sns.histplot(data=iris_data, x="sepal.width", bins=10, color="black")
plt.title("Histogram of Sepal Width")
plt.show()

sns.relplot(data=iris_data, x="sepal.length", y="sepal.width", hue="variety",
kind="scatter", palette="Set1")
plt.title("Scatter Plot of Sepal Length vs. Sepal Width")
plt.show()
```

Output:




Histogram of Sepal Width
Count
sepal.width
0
5
10
15
20
25
30
35
2.0
2.5
3.0
3.5
4.0
4.5


Scatter Plot of Sepal Length vs. Sepal Width
sepal.width
sepal.length
variety
Setosa
Versicolor
Virginica
2.0
2.5
3.0
3.5
4.0
4.5
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0